



Установка самого быстрого сервера на основе Centos 6 + Nginx + PHP5 (PHP-FPM) и MySQL

Как быстро установить сервер для сайта, причем в качестве веб сервера использовать nginx ?

В данной статье мы коротко расскажем, как установить самый быстрый хостинг с поддержкой php для вашего сайта. Для нашей тестовой лаборатории мы будем использовать такое имя server1.example.com и IP адрес 192.168.0.100. Как вы уже наверно догадались, php-fpm не найти в базовых репозиториях, по этому мы будем добавлять репозиторий Remi RPM, но для его установки нам необходим репозиторий epel.

Для этого нам необходимо установить следующие пакеты на сервер:

```
rpm -ivh http://rpms.famillecollet.com/enterprise/remi-release-6.rpm
rpm --import https://fedoraproject.org/static/0608B895.txt
rpm -ivh http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
rpm --import http://rpms.famillecollet.com/RPM-GPG-KEY-remi
```

Устанавливаем функционал приоритетов для yum:

```
yum install yum-priorities
```

Редактируем epel.repo:

```
vi /etc/yum.repos.d/epel.repo
```

Для не базовых репозиториях зададим приоритет для установки пакетов, priority=10

```
[epel]
name=Extra Packages for Enterprise Linux 6 - $basearch
#baseurl=http://download.fedoraproject.org/pub/epel/6/$basearch
mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=epel-6&arch=$basearch
failovermethod=priority
enabled=1
priority=10
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-6
[...]
```



Редактируем remi.repo:

```
vi /etc/yum.repos.d/remi.repo
```

```
[remi]
name=Les RPM de remi pour Enterprise Linux $releasever - $basearch
#baseurl=http://rpms.famillecollet.com/enterprise/$releasever/remi/$basearch/
mirrorlist=http://rpms.famillecollet.com/enterprise/$releasever/remi/mirror
enabled=1
priority=10
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-remi
failovermethod=priority
```

Устанавливаем mysql 5:

```
yum install mysql mysql-server
```

Запускаем и делаем его автозапуск:

```
chkconfig --levels 235 mysqld on
/etc/init.d/mysqld start
```

После установки запускаем скрипт конфигурации, для установки базовых параметров сервера mysql:

```
mysql_secure_installation
```

Теперь мы можем переходить к установке nginx:

```
yum install nginx
```



Запускаем и ставим в автозагрузку.

```
chkconfig --levels 235 nginx on  
/etc/init.d/nginx start
```

После установки nginx на сервер можем приступить к установке и настройке php:

```
yum install php-fpm php-cli php-mysql php-gd php-imap php-ldap php-odbc php-pear php-xml php-xmlrpc  
php-magickwand php-magpierss php-mbstring php-mcrypt php-mssql php-shout php-snmp php-soap php-  
tidy  
yum install php-pecl-apc
```

Далее открываем на редактирование /etc/php.ini и устанавливаем параметр *cgi.fix_pathinfo* равным 0.

Дополнительно к этому устанавливаем параметр *date.timezone* для того чтоб не выводились сообщения об ошибках вида:

```
[08-Aug-2011 18:07:08] PHP Warning: phpinfo(): It is not safe to rely on the system's timezone settings.  
You are *required* to use the date.timezone setting or the date_default_timezone_set() function. In case  
you used any of those methods and you are still getting this warning, you most likely misspelled the  
timezone identifier. We selected 'Europe/Berlin' for 'CEST/2.0/DST' instead in /usr/share/nginx/html/info.php  
on line 2
```

Найти значение параметра *date.timezone* можно командой:

```
cat /etc/sysconfig/clock
```

Далее настраиваем php-fpm на автозагрузку:

```
chkconfig --levels 235 php-fpm on  
/etc/init.d/php-fpm start
```

Переходим к настройке сервера NGINX:

Открываем конфигурационный файл Web-сервера Nginx:

```
vi /etc/nginx/nginx.conf
```



Одним из отличительных черт установки этого сервера является простота его конфигурационного файла, подробности по более тонкой настройке сервера можно узнать в официальной документации по адресу <http://wiki.codemongers.com/NginxFullExample> или: <http://wiki.codemongers.com/NginxFullExample2>).

Но а нам самое первое что необходимо настроить в нашем веб-сервере это `worker_processes` и `keepalive_timeout` (но это опционально или если этого требует Ваш проект).

```
[...]  
worker_processes 4;  
[...]  
    keepalive_timeout 2;  
[...]
```

Наши виртуальные хосты настраиваются в секции `server {}` в `/etc/nginx/conf.d` директории. Виртуальный хост по умолчанию находится в `/etc/nginx/conf.d/default.conf` давайте его настроим под наши нужды:

```
vi /etc/nginx/conf.d/default.conf
```

Пример моего файла настройки:

```
[...]  
server {  
    listen 80;  
    server_name _;  
    #charset koi8-r;  
    #access_log logs/host.access.log main;  
    location / {  
        root /usr/share/nginx/html;  
        index index.php index.html index.htm;  
    }  
    error_page 404 /404.html;  
    location = /404.html {  
        root /usr/share/nginx/html;
```



```
}  
  
# redirect server error pages to the static page /50x.html  
  
#  
error_page 500 502 503 504 /50x.html;  
location = /50x.html {  
    root /usr/share/nginx/html;  
}  
  
# proxy the PHP scripts to Apache listening on 127.0.0.1:80  
  
#  
#location ~ /\.php$ {  
#    proxy_pass http://127.0.0.1;  
#}  
  
# pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000  
  
#  
location ~ /\.php$ {  
    root /usr/share/nginx/html;  
    try_files $uri =404;  
    fastcgi_pass 127.0.0.1:9000;  
    fastcgi_index index.php;  
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;  
    include fastcgi_params;  
}  
  
# deny access to .htaccess files, if Apache's document root  
# concurs with nginx's one  
#  
location ~ /\.ht {  
    deny all;  
}  
  
}
```

server_name _; - Делает мой хост который вызывается по умолчанию.



В секции `location` / указывается где находится мой индексный файл. `root /usr/share/nginx/html;` - указывает на домашнюю директорию моего виртуального хоста.

Основная и самая главная секция для обработки PHP это `location ~ \.php$ {}`. Для того чтоб эта секция работала просто раскомментируйте ее.

Пожалуйста, убедитесь в том что Вы изменили параметр `fastcgi_param` на `fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name`; Т.к. в другом случае интерпритатор PHP просто не сможет найти файлы ваших скриптов (`$document_root` преобразится в `/usr/share/nginx/html` т.к. мы указали его как `document root`).

PHP-FPM слушает порт `9000` на хосте `127.0.0.1` (по умолчанию), поэтому необходимо сказать `nginx` подключаться к порту `9000` на `127.0.0.1` в строке `fastcgi_pass 127.0.0.1:9000;`. Также можно запускать PHP-FPM как Unix сокет об этом я расскажу позже.

Сохраняем конфигурационный файл и перезапускаем наш `nginx` сервис:

```
Service nginx restart
```

Теперь можно проверить работоспособность Вашего PHP на хостинге.

Настройка PHP-FPM на работу через сокет.

Редактируем файл `/etc/php-fpm.d/www.conf`

```
vi /etc/php-fpm.d/www.conf
```

и изменяем параметр `listen`.

```
[...]  
;listen = 127.0.0.1:9000  
listen = /tmp/php5-fpm.sock  
[...]
```

Теперь перезапускаем PHP-FPM:

```
Service php-fpm reload
```

Теперь необходимо у всех Ваших виртуальных хостов изменить параметр `fastcgi_pass 127.0.0.1:9000;` на `fastcgi_pass unix:/tmp/php5-fpm.sock;`:



```
vi /etc/nginx/conf.d/default.conf
```

Пример моего кода:

```
[...]  
  
location ~ /\.php$ {  
  
    root          /usr/share/nginx/html;  
  
    try_files $uri =404;  
  
    fastcgi_pass  unix:/tmp/php5-fpm.sock;  
  
    fastcgi_index index.php;  
  
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;  
  
    include      fastcgi_params;  
  
    }  
  
[...]
```

И наконец, перезапускаем nginx:

```
Service nginx reload
```

Если у Вас необходимо установить или настроить сервер, обращайтесь Office@itfb.com.ua